

14 JMX监控

Overview

JMX监控可用于监视Java应用程序的JMX计数器。

自Zabbix 2.0以来，JMX监视器以Zabbix守护进程方式运行，名为“Zabbix Java gateway”。

要检索主机上特定JMX计数器的值，Zabbix服务器查询Zabbix Java gateway。该网关又使用JMX管理API来远程查询感兴趣的应用程序。

有关Zabbix Java gateway的设置和更多详细信息，请参见各自的手册部分。

为Java应用程序启用远程JMX监视

Java应用程序不需要安装任何其它软件，但需要使用以下指定的命令行选项启动以支持远程JMX监视。

在最小情况下，如果你只想通过在本地主机上监控一个简单的Java应用程序，而不需要执行安全性，那么请使用以下选项启动它们：

```
java \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=12345 \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-Dcom.sun.management.jmxremote.ssl=false \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

这使Java可以侦听来自本地主机的端口12345上的传入JMX连接，并告知不要求身份验证或SSL。

如果要允许其它接口上的连接，请将-Djava.rmi.server.hostname参数设置为该接口的IP。

如果你希望更加严格的安全性，可以使用更多Java选项。例如，下一个示例使用更多的启动选项，并将其打开到更广泛的网络，而不仅仅是本地主机。

```
java \  
-Djava.rmi.server.hostname=192.168.3.14 \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=12345 \  
-Dcom.sun.management.jmxremote.authenticate=true \  
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-  
openjdk/management/jmxremote.password \  
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-  
openjdk/management/jmxremote.access \  
-Dcom.sun.management.jmxremote.ssl=true \  
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \  
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \  
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \  
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \  
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

这些设置中的大多数（如果不是全部）可以在/etc/java-6-openjdk/management/management.properties 或者系统上的任何文件位置）中指定。

请注意，如果你希望使用SSL 则必须通过向Java网关添加“-Djavax.net.ssl.*”选项来修改startup.sh脚本，以便知道在哪里可以找到密钥和信任存储。

查看[使用JMX监控和管理](#)的详细说明。

在Zabbix Web管理端配置JMX接口和监控项

Java网关在运行时，服务器知道在哪里找到它，而Java应用程序开始进行远程JMX监视，现在可以在Zabbix GUI中配置接口和监控项了。

配置JMX接口

首先在感兴趣的主机上创建一个JMX类型的接口：



添加JMX代理项

对于你感兴趣的每个JMX计数器，都可以添加一个JMX代理类型的监控项。如果你已经在Java应用程序上配置了身份验证，那么你还可以指定用户名和密码。

下面的屏幕截图中的关键配置 `jmx["java.lang:type=Memory", "HeapMemoryUsage.used"]`。关键包括2个参数：

- object name - 它代表MBean的对象名称
- attribute name - 一个MBean属性名称，可选的复合数据字段名称以点分隔

有关JMX监控项Key的更多详细信息，请参阅下文。



如果要监视一个“true”或“false”的布尔计数器，那么你将信息的类型指定为“Numeric unsigned” 数据类型指定为“Boolean” 服务器将分别将布尔值存储为1或0。

JMX监控项Key更详细的信息

简单属性

An MBean object name is nothing but a string which you define in your Java application. An attribute name, on the other hand, can be more complex. In case an attribute returns primitive data type (an integer, a string etc.) there is nothing to worry about, the key will look like this: MBean对象名称只不过是在Java应用程序中定义的字符串。另一方面，属性名称可能更复杂。如果一个属性返回原始数据类型（一个整数、一个字符串等），那就没有什么可担心的了，这个Key类似如下：

```
jmx[com.example:Type=Hello,weight]
```

在此示例中，对象名称为“com.example[]Type = Hello”[]属性名称为“weight”[]可能返回的值类型应为“Numeric[]float”[]

属性返回复合数据

当属性返回复合数据时将会更加复杂。例如：属性名称是“apple”[]它返回一个表示其参数的哈希，如“weight”[]“color”等。Key可能如下所示：

```
jmx[com.example:Type=Hello,apple.weight]
```

这是通过使用点符号来分割属性名称和哈希键的方式。同样的，如果一个属性返回嵌套的复合数据，这些部分由一个点分隔开：

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

关于点的问题

到现在为止还挺好。但是，如果属性名称或散列键包含点符号怎么办？ 这是一个例子：

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

这是一个问题。如何告诉Zabbix属性名称是“all.fruits”[]而不只是“全部”？ 如何区分属于名称和散列键点名称的一部分点？

在2.0.4之前Zabbix Java网关无法处理这种情况，用户留下了UNSUPPORTED项。 从2.0.4开始解决了此问题，所有你需要做的就是用一个反斜线来转义名字的一部分点：

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

同样的方法，如果你的散列键包含一个你可以回避的点：

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

其它问题

反斜杠字符也应该被转义：

```
jmx[com.example:type=Hello,c:\\documents]
```

如果对象名称或属性名称包含空格或逗号、双引号：

```
jmx["com.example:Type=Hello","fruits.apple.total weight"]
```

这就是全部了[]JMX监控快乐！

Last update: 2017/04/22 11:55 zh:manual:config:items:itemtypes:jmx_monitoring https://www.zabbix.com/documentation/3.4/zh/manual/config/items/itemtypes/jmx_monitoring

From: <https://www.zabbix.com/documentation/3.4/> - **Zabbix Documentation 3.4**

Permanent link: https://www.zabbix.com/documentation/3.4/zh/manual/config/items/itemtypes/jmx_monitoring

Last update: **2017/04/22 11:55**

